Attorney's Docket No.: 9387-2   PATENT

A·F/
IFW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re: Blaker et al.   Confirmation No.: 2529
Serial No.: 09/845,432   Group Art Unit: 2166
Filed: April 30, 2001   Examiner: Mohammad Ali
For: HASH-ORDERED DATABASES AND METHODS, SYSTEMS AND COMPUTER
        PROGRAM PRODUCTS FOR USE OF A HASH-ORDERED DATABASE

Date: September 29, 2006

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF
(PATENT APPLICATION--37 C.F.R. § 41.37)

1.   Transmitted herewith is the APPEAL BRIEF for the above-identified application, pursuant to
the Notice of Appeal filed on August 22, 2006.

2.   This application is filed on behalf of
        ☐   a small entity.

3.   Pursuant to 37 C.F.R. § 41.20(b)(2), the fee for filing the Appeal Brief is:
        ☐   small entity $250.00
        ☒   other than small entity $500.00

                                    Appeal Brief fee due $500.00
        ☒   Charge the $500.00 Appeal Brief fee to Deposit Account No. 50-0220.

        ☒   Any additional fee or refund may be charged to Deposit Account
                50-0220.

                                            Respectfully submitted,

                                            [signature]

                                            D. Scott Moore
                                            Registration No.: 42,011

Myers Bigel Sibley & Sajovec, P.A.
P. O. Box 37428
Raleigh, North Carolina 27627
Telephone: (919) 854-1400
Facsimile: (919) 854-1401
Customer No. 20792

Attorney's Docket No.: 9387-2

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re: Blaker et al.
Serial No.: 09/845,432
Filed: April 30, 2001
For: HASH-ORDERED DATABASES AND METHODS, SYSTEMS AND COMPUTER PROGRAM PRODUCTS FOR USE OF A HASH-ORDERED DATABASE

Confirmation No.: 2529
Group Art Unit: 2166
Examiner: Mohammad Ali

Date: September 29, 2006

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450
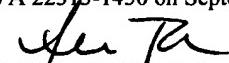
APPELLANTS' BRIEF ON APPEAL UNDER 37 C.F.R. §41.37

Sir:

This Appeal Brief is filed pursuant to the "Notice of Appeal to the Board of Patent Appeals and Interferences" filed August 22, 2006.

Real Party In Interest

The real party in interest is assignee Hifn Inc., Los Gatos, California.

Related Appeals and Interferences

Appellant is aware of no appeals or interferences that would be affected by the present appeal.

Status of Claims

Appellants appeal the rejection of Claims 1 - 13, 37, and 40 as set forth in the Final Office Action of July 25, 2006 (hereinafter "Final Action"), which as of the filing date of this Brief remain under consideration. Claims 1 - 6, 10 - 13, 37, and 40 stand rejected; Claims 7 - 9 stand objected to; and Claims 14 - 36, 38, 39, and 41 - 62 have been canceled. Appellants submit that the claims involved in the appeal are Claims 1 - 13, 37, and 40 as a reversal of the rejection of independent Claims 1, 37, and 40 is requested in the present appeal and a reversal of the rejection of dependent Claims 2 - 13 is also requested based on the reversal of the rejection of the independent claims. Accordingly, Claims 1 - 13, 37, and 40 as included in Appellants' response to the Office Action of November 3, 2005 are attached hereto as Appendix A.

Status of Amendments

No responses after final rejection have been filed in the present case.

Summary of Claimed Subject Matter

Independent Claim 1 is directed to a method of searching a database comprising generating a hash key value based on a plurality of selector values (FIG. 2; Specification page 16, lines 7 - 19), selecting an entry in the database having an address corresponding to the hash key value (block 102 of FIG. 4), wherein entries in the database include corresponding hash values, evaluating the selected entry to determine if the entry in the database corresponds to the plurality of selector values (block 104 of FIG. 4), incrementing the address corresponding to the hash key value if the selected entry does not correspond to the plurality of selector values (block 108 of FIG. 4), wherein the selecting, the evaluating and the incrementing are repeated until the hash value included in selected entry has a value which indicates that entries subsequent to the selected entry will not correspond to the plurality of selector values (block 112 of FIG. 4). (Specification, page 18, line 12 - page 19, line 24).

Independent Claim 37 is directed to a system searching a database comprising means for generating a hash key value based on a plurality of selector values (FIG. 2; Specification page 16, lines 7 - 19), means for selecting an entry in the database having an address corresponding to the hash key value (block 102 of FIG. 4), wherein entries in the database include corresponding

hash values, means for evaluating the selected entry to determine if the entry in the database corresponds to the plurality of selector values, means for incrementing the address corresponding to the hash key value if the selected entry does not correspond to the plurality of selector values (block 104 of FIG. 4), and means for repeatedly selecting, evaluating and incrementing until the selected entry has a null value or the hash value included in selected entry has a value other than the hash key value (block 112 of FIG. 4). (Specification, page 18, line 12 - page 19, line 24). The IPSec processor 20 and hash key generator 26 of FIG. 1 provide structure for the means for generating. The IPSec processor provides structure for the means for selecting, means for evaluating, means for incrementing, and means for repeatedly selecting, evaluating, and incrementing.

Independent Claim 40 is directed to a computer program product for searching a database, comprising a computer-readable storage medium having computer-readable program code embodied therein (Specification, page 11, line 5 - page 12, line 2). The computer readable program code comprises computer-readable program code which generates a hash key value based on a plurality of selector values (FIG. 2; Specification page 16, lines 7 - 19), computer-readable program code which selects an entry in the database having an address corresponding to the hash key value (block 102 of FIG. 4), wherein entries in the database include corresponding hash values, computer-readable program code which evaluates the selected entry to determine if the entry in the database corresponds to the plurality of selector values, computer-readable program code which increments the address corresponding to the hash key value if the selected entry does not correspond to the plurality of selector values (block 104 of FIG. 4), and computer-readable program code which repeatedly selects, evaluates and increments until the selected entry has a null value or the hash value included in selected entry has a value other than the hash key value (block 112 of FIG. 4). (Specification, page 18, line 12 - page 19, line 24).

Grounds of Rejection to be Reviewed on Appeal

Independent Claims 1, 37, and 40 stand rejected under 35 U.S.C. §102(b) as being anticipated by U. S. Patent Application No. 5,511,190 to Sharma et al. (hereinafter "Sharma").

Argument

I.   Introduction to 35 U.S.C. §102 Analysis

Under 35 U.S.C. § 102, "a claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." M.P.E.P. § 2131 (quoting *Verdegaal Bros. v. Union Oil Co.*, 814 F.2d 628, 631, 2 U.S.P.Q.2d 1051, 1053 (Fed. Cir. 1987)). "Anticipation under 35 U.S.C. § 102 requires the disclosure in a single piece of prior art of each and every limitation of a claimed invention." *Apple Computer Inc. v. Articulate Sys. Inc.*, 57 U.S.P.Q.2d 1057, 1061 (Fed. Cir. 2000). "The fact that a certain result or characteristic may occur or be present in the prior art is not sufficient to establish the inherency of that result or characteristic. To establish inherency, the extrinsic evidence 'must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.'" M.P.E.P. § 2112 (citations omitted).

A finding of anticipation further requires that there must be no difference between the claimed invention and the disclosure of the cited reference as viewed by one of ordinary skill in the art. *See Scripps Clinic & Research Foundation v. Genentech Inc.*, 927 F.2d 1565, 1576, 18 U.S.P.Q.2d 1001, 1010 (Fed. Cir. 1991). In particular, the Court of Appeals for the Federal Circuit held that a finding of anticipation requires absolute identity for each and every element set forth in the claimed invention. *See Trintec Indus. Inc. v. Top-U.S.A. Corp.*, 63 U.S.P.Q.2d 1597 (Fed. Cir. 2002). Additionally, the cited prior art reference must be enabling, thereby placing the allegedly disclosed matter in the possession of the public. *In re Brown*, 329 F.2d 1006, 1011, 141 U.S.P.Q. 245, 249 (C.C.P.A. 1964). Thus, the prior art reference must adequately describe the claimed invention so that a person of ordinary skill in the art could make and use the invention.

Appellants respectfully submits that the pending independent claims are patentable over the cited references for at least the reason that the cited references do not disclose or suggest, either alone or in combination, each of the recitations of the independent claims. The patentability of the pending claims is discussed in detail hereinafter.

A.   **Independent Claims 1, 37, and 40 are Patentable**

Independent Claims 1, 37, and 40 stand rejected under 35 U.S.C. §102(b) as being anticipated by Sharma. (Final Action, page 4). Independent Claim 1 is directed to a method of responding to a customer communication that recites, in part:

> ...
>> generating a hash key value based on a plurality of selector values;
>> selecting an entry in the database having an address corresponding to the hash key value, **wherein entries in the database include corresponding hash values;**
>> evaluating the selected entry to determine if the entry in the database corresponds to the plurality of selector values;
>> incrementing the address corresponding to the hash key value if the selected entry does not correspond to the plurality of selector values;
>> wherein the selecting, the evaluating and the incrementing are repeated until the hash value included in selected entry has a value which indicates that entries subsequent to the selected entry will not correspond to the plurality of selector values. (Emphasis added).

Independent Claims 37 and 40 include similar recitations. Thus, according to the independent claims, a hash key value is generated based on a plurality of selector values. An entry is selected in a database that has an address corresponding to the hash key value. Moreover, the entries in the database include corresponding hash values.

The Final Action cites a passage at column 2, lines 63 - 67 of Sharma as disclosing these recitations of the present invention. (Final Action, pages 4 and 5). This passage from the Summary of the Invention section refers to reading the entries in table T1 212 in memory 116. According to independent Claims 1, 37, and 40, a database entry is selected that has an address corresponding to the hash key value. As is known to those skilled in the art, a hash key value is provided to a hash function and the output of the hash function is used to identify a location or address in a data structure where particular data or information are stored. Sharma discloses a hash function 210 that is applied to the unique values in the group columns GC 252 of the table T1 212 to generate an index for hash table 216. (Sharma, col. 8, lines 5 - 10). Thus, the information in table T1 212 is analogous to hash key values.

The output values or addresses from the hash function 210 of Sharma are stored in the hash table 216. These addresses in the hash table 216 identify the location of various entries in the group table 218. As highlighted above, however, independent Claim 1 recites "selecting an entry in the database having an address corresponding to the hash key value, **wherein entries in the database include corresponding hash values.**" Appellants acknowledge that the group table 218 is a database from which entries may be selected corresponding to a hash key value where the unique values in table T1 212 are interpreted as hash key values. Appellants submit, however, that unlike the recitations of independent Claims 1, 37, and 40, **the entries in the group table 218 do not include corresponding hash values**. As shown in FIG. 2, the group table 218 includes three columns corresponding to dname, sum_salary, and count. None of these columns corresponds to the hash value of an entry in the group table 218, e.g., sum_salary is not generated by hashing dname or vice versa. It appears that the group table 218 merely contains summarized/aggregated data based on the table T1 212, *see, e.g.*, Sharma, col. 6, line 56 - col. 7, line 16).

In response to these arguments, the Final Action cites various tables and flags disclosed in Sharma (Final Action, page 3), but does not point out where it is described that the group table 218 includes both entries and hash values of those entries in the same table. Moreover, if the Final Action is not citing group table 218 as disclosing the database entries recited in independent Claims 1, 37, and 40, then Appellants submit that none of the other tables disclosed in Sharma include entries and corresponding hash values for those entries in the same table. Appellants wish to point out that independent Claims 1, 37, and 40 recite that the database **entry** includes a corresponding hash value. Thus, if the Final Action is rejecting the pending independent claims under the theory that the database server 102 includes memory where database entries and hash values associated therewith are stored across various tables/data structures, then this is insufficient because independent Claims 1, 37, and 40 require a database entry to include the hash value of the entry data as part of the entry.

For at least the foregoing reasons, Appellants submits that independent Claims 1, 37, and 40 are patentable over the cited reference and that dependent Claims 2 - 13 are patentable at least by virtue of their depending from an allowable claim. Accordingly, Appellants respectfully request that the rejection of Claims 1 - 13, 37, and 40 be reversed based on the failure of the

Examiner to establish a prima facie case of anticipation under 35 U.S.C. §102 for at least these reasons.

B. Dependent Claims 7 - 9 are Separately Patentable

Dependent Claims 7 - 9 stand objected to as being dependent on a rejected base claim. (Final Action, page 8). Appellants note that Claims 7 - 9 are patentable for the reasons discussed above in Section IA and are also separately patentable for the reasons discussed in the Final Action on page 8 in the section entitled "Allowable Subject Matter."

II. Conclusion

In summary, Appellants respectfully submit that, with respect to Claims 1 - 13, 37, and 40, the cited reference does not teach all of the recitations of the claims for at least the reasons discussed above. Accordingly, Appellants respectfully request reversal of the rejection of Claims 1 - 13, 37, and 40 based on the cited references.

Respectfully submitted,
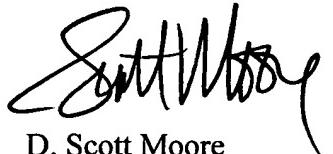
D. Scott Moore
Registration No. 42,011

Myers Bigel Sibley & Sajovec, P.A.
P. O. Box 37428
Raleigh, North Carolina 27627
Telephone: (919) 854-1400
Facsimile: (919) 854-1401
Customer No. 20792

APPENDIX A

1.   (Original) A method of searching a database, the method comprising:

generating a hash key value based on a plurality of selector values;

selecting an entry in the database having an address corresponding to the hash key value, wherein entries in the database include corresponding hash values;

evaluating the selected entry to determine if the entry in the database corresponds to the plurality of selector values;

incrementing the address corresponding to the hash key value if the selected entry does not correspond to the plurality of selector values;

wherein the selecting, the evaluating and the incrementing are repeated until the hash value included in selected entry has a value which indicates that entries subsequent to the selected entry will not correspond to the plurality of selector values.

2.   (Original) A method according to Claim 1, wherein the selecting, the evaluating and the incrementing are repeated until an entry corresponding to the plurality of selector values is reached or until the hash value included in the selected entry has a value which indicates that entries subsequent to the selected entry will not correspond to the plurality of selector values.

3.   (Original) A method according to Claim 1, wherein the selecting, the evaluating and the incrementing are repeated until the selected entry is a null entry.

4.   (Original) A method according to Claim 1, wherein the selecting, the evaluating and the incrementing are repeated until the selected entry has a hash value greater than the hash key value.

5.   (Original) The method of Claim 2, further comprising:

providing the selected entry if the selected entry corresponds to the plurality of selector values; and

providing an indicator of failure of the search if the selected entry includes a hash value other than the hash key value or the selected entry has a null value.

6.   (Original) The method of Claim 1, wherein generating a hash key value based on a plurality of selector values comprises encrypting the selector values to provide the hash key value.

7.   (Original) The method of Claim 6, wherein encrypting the selector values to provide the hash key value comprises:

grouping the plurality of selector values into blocks having a predefined number of bits;

padding the blocks of grouped selector values to the predefined number of bits;

encrypting the padded blocks; and

truncating the encrypted padded blocks to a number of bits in the hash key value to provide the hash key value.

8.   (Original) The method of Claim 7, wherein encrypting the padded blocks comprises encrypting the padded blocks using Cipher-Block-Chaining encryption mode of Data Encryption Standard (DES-CBC) encryption.

9.   (Original) The method of Claim 8, wherein the database comprises an Internet Protocol Security (IPSec) security association database, the plurality of selector values comprise IPSec selector fields and the predefined number of bits comprises 64 bits.

10.   (Original) The method of Claim 1, wherein the database comprises an Internet Protocol Security (IPSec) security association database and the plurality of selector values comprise IPSec selector fields.

11.   (Original) The method of Claim 10, wherein the database has a size of about four times a maximum number of supported security associations.

12. (Original) The method of Claim 1, wherein the database is contained in a circular memory and wherein incrementing the address comprises:

incrementing the address to a next consecutive address if the address is less than a maximum address of the circular memory; and

setting the address to a first address of the circular memory if the address is equal to the maximum address of the circular memory.

13. (Original) The method of Claim 12, wherein the selecting, the evaluating and the incrementing are repeated until a hash value of the selected entry is less than a hash value of a previous selected entry and the hash value of the selected entry is greater than the hash key value.

14. - 36. (Canceled)

37. (Original) A system searching a database, comprising:

means for generating a hash key value based on a plurality of selector values;

means for selecting an entry in the database having an address corresponding to the hash key value, wherein entries in the database include corresponding hash values;

means for evaluating the selected entry to determine if the entry in the database corresponds to the plurality of selector values;

means for incrementing the address corresponding to the hash key value if the selected entry does not correspond to the plurality of selector values;

means for repeatedly selecting, evaluating and incrementing until the selected entry has a null value or the hash value included in selected entry has a value other than the hash key value.

38. - 39. (Canceled)

40. (Original) A computer program product for searching a database, comprising:

a computer-readable storage medium having computer-readable program code embodied therein, the computer readable program code comprising:

computer-readable program code which generates a hash key value based on a plurality of selector values;

computer-readable program code which selects an entry in the database having an address corresponding to the hash key value, wherein entries in the database include corresponding hash values;

computer-readable program code which evaluates the selected entry to determine if the entry in the database corresponds to the plurality of selector values;

computer-readable program code which increments the address corresponding to the hash key value if the selected entry does not correspond to the plurality of selector values;

computer-readable program code which repeatedly selects, evaluates and increments until the selected entry has a null value or the hash value included in selected entry has a value other than the hash key value.

41. - 62. (Canceled)

APPENDIX B – EVIDENCE APPENDIX

None

APPENDIX C – RELATED PROCEEDINGS APPENDIX

None.